



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사학위논문

듀얼 어텐션 메커니즘 기반 Bidirectional
Encoder-Decoder Recurrent Neural Network를
활용한 다변량 시계열 예측

Bidirectional Encoder-Decoder with Dual-Stage Attention for
Multivariate Time-Series Prediction

2019 년 8 월

서울대학교 대학원

협동과정 기술경영경제정책전공

이 형 욱

듀얼 어텐션 메커니즘 기반 Bidirectional Encoder-Decoder Recurrent Neural Network 를 활용한 다변량 시계열 예측

Bidirectional Encoder-Decoder with Dual-Stage Attention
for Multivariate Time-Series Prediction

지도교수 조 성 준

이 논문을 공학석사 학위논문으로 제출함

2019 년 8 월

서울대학교 대학원

협동과정 기술경영경제정책전공

이 형 욱

이형욱의 공학석사 학위논문을 인준함

2019 년 6 월

위 원 장	<u> 구 윤 모 </u>	(인)
-------	----------------------------------	-----

부위원장	<u> 조 성 준 </u>	(인)
------	----------------------------------	-----

위 원	<u> 윤 명 환 </u>	(인)
-----	----------------------------------	-----

초록

자연어처리에 주로 사용되는 RNN 계열의 모델들은 순차적인 데이터를 다루는 데 적합하기 때문에 시계열 데이터 분석에도 다양하게 활용되고 있다. RNN 계열의 모델들이 가지고 있는 취약점은 기울기 값의 소실이라는 문제이다. 이를 해결하기 위해 배치 정규화 같은 기법들이 사용되고 있지만, 시계열 데이터의 경우는 과거에서부터 이어져 오는 추세를 잃어버릴 수 있다. 그리고 적절한 외생변수들을 선택하는 사항도 고려되어야 한다. 본 연구에서는 위에서 언급한 문제들을 해결하기 위해 듀얼 어텐션 메커니즘을 기반으로 하는 양방향 Encoder-Decoder LSTM 모델을 제안한다. Encoder에서 작동하는 어텐션 메커니즘은 이전 단계의 양방향 LSTM에서 전달받은 hidden state와 cell state를 참조하여 예측에 도움이 되는 외생변수들을 파악한다. Decoder는 시계열적인 특성을 반영한 모델 구조로 되어 있다. 먼저 타겟변수의 과거 값들과 추세를 나타내는 통계치를 입력으로 받아서 양방향 LSTM을 통해 학습하고, 지정한 시간 간격까지 매번 hidden state를 업데이트시킨다. Decoder에서의 어텐션은 업데이트된 hidden state와 Encoder에서 나온 출력값을 연결하여 입력으로 받는다. 따라서 예측에 적합한 외생변수의 시점을 파악하는 것뿐만 아니라 타겟변수의 긴 추세를 반영할 수 있다. 모델 평가에 사용된 데이터는 약 4년 치의 KODEX 200(ETF)과 KODEX 200에 포함된 회사들의 5분 단위 개별 주가 거래 데이터이다.

주요어: Encoder-Decoder, Attention Mechanism, Recurrent Neural Network, Time-Series Prediction, Finance

학번: 2017-22550

목차

초록	i
목차	iii
표 목차	iv
그림 목차	v
제 1 장 서론	1
제 2 장 관련 연구	4
2.1 시계열 분석에 대한 선행 연구	4
2.2 Encoder-Decoder 모델에 대한 선행 연구	6
2.3 어텐션 메커니즘에 대한 선행 연구	9
제 3 장 제안하는 방법	13
3.1 제안하는 Encoder-Decoder의 구조	13
3.2 Encoder의 내부구조	15
3.3 Decoder의 내부구조	17
제 4 장 실험 결과	20
4.1 데이터 설명	20
4.2 데이터 전처리 및 학습 방법	22

4.3	성능 평가 및 비교	25
4.4	어텐션 가중치 분석	28
제 5 장	결론	35
	참고문헌	37
	Abstract	42

표 목차

표 4.1	사용한 요약 통계량 변수	23
표 4.2	성능 평가 결과표	26
표 4.3	타겟변수만을 이용한 예측 모델의 성능	27
표 4.4	상관계수 상위 10개 종목	30

그림 목차

그림 2.1	LSTM의 구조와 수식[16]	5
그림 2.2	기본적인 Encoder-Decoder 구조[8]	6
그림 2.3	시계열 분류 Encoder-Decoder 모델[28]	8
그림 2.4	어텐션 메커니즘 방법[2]	9
그림 3.1	제안하는 모델의 구조	14
그림 3.2	디코더의 구조	17
그림 4.1	상위 8개 구성종목의 구성비중	20
그림 4.2	데이터의 예시	21
그림 4.3	Bollinger Bands의 사용 예시	24
그림 4.4	타겟변수만을 활용한 LSTM예측모델	27
그림 4.5	어텐션 가중치와 랜덤 노이즈의 비교	28
그림 4.6	KODEX 200 모든 구성종목의 어텐션 가중치	29
그림 4.7	어텐션 가중치가 높은 상위 5개 종목(20,653번째 예측값)	31
그림 4.10	어텐션 가중치가 낮은 하위 7개 종목(20,653번째 예측값)	33

제 1 장 서론

딥러닝 알고리즘이 발달하면서 시계열 데이터를 다루는 연구도 여러 분야에서 진행되고 있다. 대표적인 분야는 주식시장의 거래 데이터[31], 날씨 데이터[5], 공정 센서 데이터[22] 등이 있다. 주로 시계열 데이터를 다루는 알고리즘에는 자연어처리에서 사용되는 Recurrent Neural Network(RNN) 계열의 모델들이 사용된다. 한 문장 안에 존재하는 단어들을 순차적으로 학습하는 방법이 시계열 데이터에 적용하기에도 용이하기 때문이다. 따라서 자연어처리 분야에서 두각을 나타내는 모델들을 이용하여 시계열 분석이나 예측에 활용하고 있다.

하지만, 최근 자연어처리 분야의 연구 방향을 살펴보면 LSTM, GRU 같은 대표적인 RNN 계열의 모델들이 외면을 받고 있다. RNN 모델을 사용하지 않고, Multi-head 어텐션 메커니즘을 활용한 Transformer[29], BERT[10] 같은 모델들이 자연어처리의 여러 분야에서 두각을 나타내고 있기 때문이다. Multi-head 어텐션 메커니즘을 시계열 데이터에 적용한 연구들도[26, 20] 있다. 그러나 시계열 데이터는 자연어와 다른 특성이 있기에 적용하는 데 한계가 있다. 단어 같은 경우에는 word2vec[23]과 같은 알고리즘을 사용해서 의미를 가지는 벡터로 임베딩을 할 수 있지만, 시계열 데이터는 각 데이터를 임베딩하여 유의미한 벡터를 만들기 어렵기 때문이다. 또한, Encoder-Decoder 구조로 되어 있는 모델은 입력으로 받은 데이터의 길이가 길면 성능이 급격하게 감소하는 단점이 있다[15]. 이는 시계열 분석에 있어서 큰 어려움이 된다. 시계열 분석에서 예측은 타겟변수와 외생변수들의 상대적으로 긴 추세와 움직임을 바탕으로 이루어지기 때문이다.

위와 같은 이유로 시계열 데이터가 가지고 있는 특성을 반영한 모델이 연구되어

야 한다. 임베딩된 벡터를 기반으로 단순히 어텐션 메커니즘을 활용한 학습 방법은 시계열 데이터에 적용하기 어렵다. 또한, 모델은 시계열 데이터의 추세를 반영할 수 있어야 좋은 성능을 기대할 수 있다. 시계열적인 특성을 반영하기 위해서는 기존 RNN 계열의 모델을 활용한 학습 방법에도 개선점이 필요하다. RNN 계열의 모델들이 가진 치명적인 결함은 기울기 값의 소실 (Vanishing Gradient)이다[3]. 기울기 값의 소실이란 모델이 깊거나 길면, 오랜 시간 전의 데이터에 관한 정보를 잃어버리는 현상이다. 비록 LSTM이나 GRU 같은 모델들은 기울기 값의 소실 문제를 완화해주고[16], 드롭아웃[27], ReLU[24], 배치 정규화(Batch Normalization)[17] 등과 같은 기법들이 문제를 해결해 줄 수 있지만, 시계열 데이터에 그대로 적용을 하면 또 다른 문제를 야기할 수 있다. 예를 들면, 배치 정규화의 경우 데이터에서 뽑힌 배치에 대해서 순차적으로 학습이 진행되기 때문에 시계열의 과거에서부터 이어져 오는 추세를 반영할 수 없다.

따라서 본 연구에서는 시계열 데이터의 특성을 반영하여 듀얼 어텐션 메커니즘[25]과 LSTM을 기반으로 하는 새로운 Encoder-Decoder 구조의 모델을 제안한다. 제안하는 모델의 구조에 대한 자세한 설명은 제 3장에서 살펴볼 수 있다. 듀얼 어텐션 메커니즘은 Qin et al.[25]이 제안한 방법이다. 인코더와 디코더에 각각 어텐션이 적용되기 때문에 듀얼 어텐션 메커니즘이라고 불린다. 인코더에서의 어텐션 메커니즘은 다음 예측값에 관련있는 외생변수들을 판단한다[25]. Decoder에서 작동하는 어텐션 메커니즘은 시간적인 방법으로, 모든 시간간격에 대해서 관련성이 있는 변수들에 대한 인코더의 hidden states를 선택한다[25]. Qin et al.[25]이 제안한 DA-RNN 모델은 Nonlinear Autoregressive Exogenous (NARX) 모델이다. NARX 모델을 수식화하면 아래와 같다.

$$y_t = F(y_{t-1}, y_{t-2}, y_{t-3}, \dots, u_t, u_{t-1}, u_{t-2}, u_{t-3}, \dots) + \epsilon_t \quad (1.1)$$

수식(1.1)에서 y 는 예측하고자 하는 타겟변수, u 는 y 를 예측하는데 도움이 되는 외

생변수, 그리고 함수 F 는 신경망같은 비선형 함수이다. 수식(1.1)에서 보다시피 y_t 를 예측하는데 외생변수 u 의 시점이 u_t 까지 인풋으로 들어가있다. 하지만 주가를 예측하는 테스크의 경우, 현재의 가격 y_t 를 예측하는데 외생변수의 시점이 u_t 까지 반영된다면 현실적이지 않은 예측이라고 볼 수 있다. 그러므로, 본 연구에서는 인풋으로 사용하는 모든 외생변수들의 시점을 u_{t-1} 까지만 사용하는 진정한 의미의 시계열 예측모델을 제안한다.

본 논문은 5 장으로 구성된다. 제 2장에서는 시계열 분석, Encoder-Decoder 구조의 모델, 그리고 어텐션 메카니즘에 대한선행 연구를 살펴본다. 제 3장에서는 본 연구에서 제안하는 모델을 소개한다. 제 4장에서는 실험에 사용되는 데이터 그리고 데이터 전처리 과정과 학습방법을 살펴본다. 마지막으로 제 5장에서는 결론과 향후 연구방향을 제시한다.

제 2 장 관련 연구

2.1 시계열 분석에 대한 선행 연구

먼저 시계열 분석에 관한 선행 연구를 살펴보면 다음과 같다. 잘 알려진 전통적인 시계열 분석 방법론 autoregressive moving model (ARMA)[30] 그리고 ARMA의 변형 모델인 ARIMA[1] 같은 모델들이 다양한 실생활 데이터를 이용하여 효율성을 입증하였지만, 데이터가 비선형적인 관계를 가지고 있는 경우 모델링을 할 수 없다는 단점이 있다. 이 같은 문제를 해결하기 위해서 nonlinear autoregressive exogenous model (NARX)이 연구되었다[21, 13]. 비선형적인 관계를 모델링하기 위한 연구는 꾸준히 진행되어왔고, 다양한 방법이 제시되었다. 커널 방법론을 접목한 NARX 모델을 이용한 시계열 예측[6], 앙상블 방법을 이용한 연구[4], 그리고 가우시안 프로세스를 사용한 연구[12]등이 있지만, 이러한 방법론들의 결점은 데이터가 가지고 있는 비선형적인 관계를 미리 정의하기 때문에 실제 관계를 제대로 모델링 하지 못 할 수도 있다[25].

심층 신경망 방법론들이 주목을 받으면서 시계열 분석에도 신경망 모델들이 적용되었다. RNN 모델의 경우 순차적인 모델링에 특화되어있고, 비선형 관계를 모델링하는데 효과적이기 때문에 많은 주목을 받았다. Gao and Er[13]은 fuzzy neural network(FNN) 모델과 recurrent FNN 모델을 이용한 NARX 시계열 분석 방법론을 제안하였고, 신경망 모델이 시계열 분석에도 뛰어난 성과를 나타냄을 입증하였다. Diaconescu[11]는 RNN 기반의 NARX 모델이 노이즈나 랜덤성이 강한 chaotic time series 분석에도 가능성이 있음을 밝혔다. 그러나 기본적인 RNN 모델은 기울기 값의 소실이라는 문제를 가지고 있다[15]. 인풋의 길이와 스텝이 증가할수록 과거의 기울기 값의 영향력이 작아지는

문제로, 시계열 분석에서는 치명적인 결함이다. 따라서 이를 해결하기 위해 다양한 종류의 RNN 모델과 기법들이 등장하였다. 대표적인 RNN 모델은 LSTM[16]과 GRU[7]가 있고, 두 모델 간에 성능 차이는 거의 없다[9]. 본 연구에서는 개선된 LSTM 모델인 bidirectional LSTM을 사용하였으며, 이에 대한 자세한 설명은 제 3장에 있다. 아래 그림 2.1은 LSTM block의 구조와 수식이다.

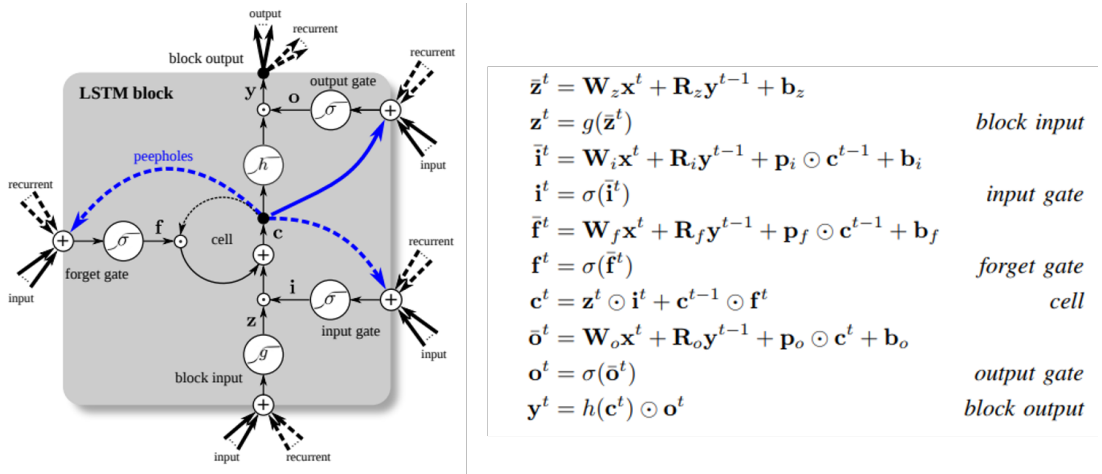


Figure 2.1: LSTM의 구조와 수식[16]

기존 RNN 구조와 다르게 LSTM은 이전 스텝의 정보를 어느정도 가져갈 지 학습하는 forget gate가 있다. 따라서 오차역전파(backpropagation)를 계산하는 과정에서 이전 단계의 기울기값이 소멸되지 않기 때문에 기울기 값의 소실 문제에 대응이 가능하다. 이 밖에도, 드롭아웃[27], ReLU[24], 배치정규화[17]처럼 기울기 값의 소실 문제를 완화시켜주는 기법들이 있다. 본 연구에서는 ReLU 활성화 함수를 사용하였으며, 배치정규화는 시계열 데이터의 특성을 반영하여 적용하였다.

2.2 Encoder-Decoder 모델에 대한 선행 연구

기계번역[2]에서 뛰어난 성능을 보인 RNN기반의 Encoder-Decoder 구조의 모델은 음성인식[14], 그리고 영상처리[18]등 여러 분야에서 활용되고 있다. Encoder-Decoder 구조는 입력 데이터를 요약하여 정보를 함축하고 있는 context vector로 표현하는 인코더와 전달받은 정보를 출력 데이터로 처리하는 디코더로 구성된다. 이 처럼 모델의 구조 안에서 인코더와 디코더의 역할이 다르지만, 학습할 때는 처음부터 끝까지 한 번에 이루어지는 end-to-end 모델이다. RNN을 기반으로 하므로 주로 텍스트 데이터를 다루는 모델이고, 연관된 연속의 데이터를 학습하는 sequence-to-sequence 학습 구조로 되어있다. 따라서 시계열 데이터를 분석하는데 적합한 모델이다. 다만, 데이터를 입력할 때 고정된 길이의 벡터로 입력을 해야 하기 때문에 사용자가 지정한 시간 간격으로 나눠서 인코딩을 해야 한다. 아래 그림 2.2는 Cho et al.[8]이 제안한 기본적인 RNN Encoder-Decoder 모델의 구조이다.

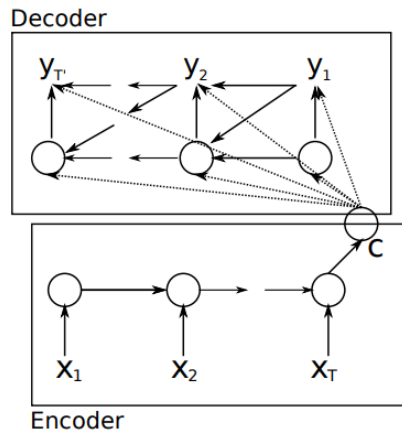


Figure 2.2: 기본적인 Encoder-Decoder 구조[8]

기본적인 RNN Encoder-Decoder 모델의 구조는 위의 그림과 같이 두개의 RNN 모델로 구성되어있다. Cho et al.[8]이 제안한 모델은 기계번역을 수행하기 위해 고

안된 구조로, 인코더에 입력되는 변수 x_T 의 길이와 디코더에서 출력되는 변수 y_T 의 길이가 같아야한다. 인코더에서는 입력변수 x 가 그림과 같이 순차적으로 정보를 전달 하면서 RNN의 hidden state가 업데이트된다. 문장이 끝나는 x_t 까지 업데이트가 되면 C (context vector)는 전체 인풋에 대한 요약정보를 표현하는 벡터가 된다. 디코더는 출력변수 y 를 생성하는 또 다른 RNN 모델이고, hidden state와 context vector를 기반으로 계산이 된다. 디코더의 hidden state는 다음과 같이 구해진다.

$$h_{<t>} = f(h_{<t-1>}, y_{t-1}, c) \quad (2.1)$$

위 식에서 $f()$ 는 tanh나 ReLU같은 비선형 활성화 함수이다. 확률적인 관점에서 모델은 y 를 생성하는 조건부확률 $p(y_1, \dots, y_T | x_1, \dots, x_T)$ 를 최대화하는 방향으로 학습이 된다[8]. 따라서 모델의 손실함수는 아래와 같다. θ 는 모델의 매개변수이다.

$$\max_{\theta} \frac{1}{N} \sum_{n=1}^N \log p_{\theta}(y_n | x_n) \quad (2.2)$$

위에서 설명한 RNN Encoder-Decoder 모델은 기본적인 Encoder-Decoder의 구조이고, 인코더와 디코더의 내부구조를 변경하여 다양한 응용모델을 만들 수 있다. 일반적으로 Encoder-Decoder 구조는 기계번역이나 자연어 분석에 많이 사용되지만, 시계열 데이터를 활용한 연구도 꾸준히 진행되고 있다. Tang et al.[28]은 LSTM 기반의 어텐션 메커니즘을 활용한 시계열 분류 Encoder-Decoder 모델을 제안하였다. 기계번역 테스트에서 디코더의 출력이 입력으로 받은 문장의 번역문장이라면, Tang et al.[28]이 제안한 모델에서는 시계열을 분류하는데 중요한 변수들을 출력한다. 아래의 그림 2.3은 Tang et al.[28]이 제안한 시계열 분류 모델이다. 어둡게 칠해진 정사각형은 디코더단에서 사용된 RNN 모델이고, 디코더에서 출력으로 나온 y_1, \dots, y_L 변수들을 완전연결층과

출력층에 연결하여 분류하도록 구성하였다.

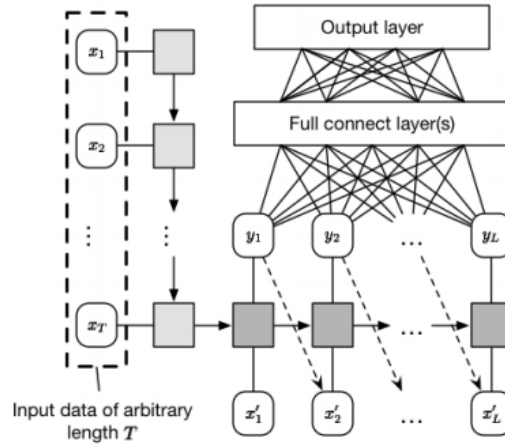


Figure 2.3: 시계열 분류 Encoder-Decoder 모델[28]

Qin et al.[25]이 제안한 모델은 LSTM 기반의 Encoder-Decoder NARX 시계열 예측 모델이다. 목차 2.3에서 서술할 듀얼 어텐션 메커니즘을 접목하여 모델의 성능과 해석력을 개선하였다. Qin et al.[25]은 모델의 성능평가로 대표적인 시계열 데이터인 온도데이터(실내온도)와 주식데이터(NASDAQ100)를 사용하였다. Qin et al.[25]은 제안하는 모델과 전통적인 시계열 예측 모델인 ARIMA와 RNN등 여러 모델과 비교하여 성능우위를 보였다. Liang et al.[20]은 Transformer[29]모델에서 제안한 Multi-level 어텐션 메커니즘의 개념을 활용하여서 LSTM 기반의 Encoder-Decoder 시계열 예측 모델을 제안하였다. Liang et al.[20]이 제안한 모델 GeoMAN은 지형공간적인 정보를 담고 있는 시계열 데이터(수질, 공기청청도)를 사용하였다. Qin et al.[25]과 연구들 모두 외생변수가 10개 이상인 시계열 데이터를 다루는 Encoder-Decoder 구조의 모델을 제안하였고, 좋은 성능을 보였다. 그러므로 RNN 기반의 Encoder-Decoder 구조가 다변량 시계열 데이터 분석에 적합한 모델이라고 판단할 수 있다.

2.3 어텐션 메커니즘에 대한 선행 연구

Cho et al.[8]이 제안한 기본적인 RNN 기반의 Encoder-Decoder 구조는 크게 두 가지의 문제점을 가지고 있다. 처음 문제는 입력 시퀀스의 길이가 길면 모델의 성능이 급격하게 감소하는 문제이고, 그 다음 문제는 모델의 성능이 context vector에 크게 좌우된다는 점이다[7]. 위와 같은 문제를 해결하기 위한 기법이 어텐션 메커니즘이다. Bahdanau et al.[2]이 제안한 어텐션 메커니즘은 모델이 테스트를 풀 때 어느 부분에 집중해야 하는지 판별하는 기법이다. 어텐션 메커니즘을 사용함으로써, 모든 입력값에 대해 동등하게 기울기 값을 가지지 않으므로, 입력 시퀀스의 길이가 길어서 생기는 문제에 대응을 할 수 있다.

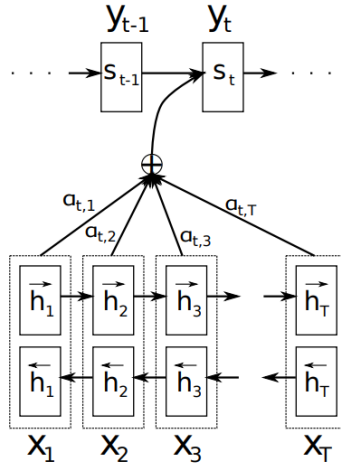


Figure 2.4: 어텐션 메커니즘 방법[2]

위 그림 2.4, Bahdanau et al.[2]가 제안한 어텐션 메커니즘을 자세히 설명하면 다음과 같다. h_i 는 인코더단에서 사용된 RNN모델의 hidden states이고, 그림 2.4에서는 bi-directional RNN 모델[schuster and paliwal, 1997]이 사용되었기 때문에 양방향의 hidden states가 존재한다. 디코더단에서 사용된 RNN 모델의 hidden states는 s_i 이고,

다음과 같이 계산된다.

$$s_i = f(s_{i-1}, y_{i-1}, c_i) \quad (2.3)$$

그리고 어텐션 메커니즘의 가중치 $a_{i,j}$ 와 context vector c_i 는 다음과 같이 계산이 된다.

$$c_i = \sum_{j=1}^{T_x} a_{i,j} h_j \quad (2.4)$$

$$a_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k=1}^{T_x} \exp(e_{i,k})} \quad (2.5)$$

$$e_{i,j} = a(s_{i-1}, h_j) \quad (2.6)$$

위 식 2.5에서 $e_{i,j}$ 는 스칼라값으로, 직전 단계의 hidden state인 s_{i-1} 과 인코더의 j번째 hidden state의 유사한 정도를 나타내는 값이다. Bahdanau et al.[2]가 alignment model이라고 명명한 a 는 s_{i-1} 와 h_j 의 유사도를 계산하는 모델로 유사도를 잘 계산할 수 있으면 변형이 가능하다. 식 2.4는 계산된 유사도 점수 $e_{i,j}$ 를 softmax 함수를 적용하여 모든 점수의 합이 1이 되도록 하는 식이다. 따라서 s_{i-1} 과 h_j 의 유사도가 높다면, 어텐션 메커니즘의 가중치 $a_{i,j}$ 가 높게 나타나게 된다. 식 2.4에서 context vector c_i 는 계산된 어텐션 점수와 인코더의 각 hidden state의 가중합으로 계산된다. 인코더에 모든 hidden states에 대해서 가중합이 계산되기 때문에 입력값의 길이가 길면 성능이 저하 되는 문제에 대응할 수 있다. 또한, context vector에 따라서 성능이 크게 좌우되는 현상도 해결할 수가 있다. 어텐션 메커니즘은 Encoder-Decoder구조를 가진 모델 뿐만 아니라 multi-head 어텐션 메커니즘[29]처럼 변형되어서 다양하게 쓰이는 기법이다. 본 연구에서는 시계열 데이터에 적합한 Qin et al.[25]이 제안한 듀얼 어텐션 메커니즘을 활용하였다.

Qin et al.[25]이 제안한 듀얼 어텐션 메커니즘은 인코더단과 디코더단에 어텐션

이 각각 적용되는 방법이다. 인코더에서 사용되는 어텐션 메커니즘은 input attention layer로 여러 외생변수들 중에서 타겟변수와 관련이 깊은 변수들을 선택하여서 가중치를 주는 방법이다[25]. Bahdanau et al.[2]이 제안한 어텐션 메커니즘과 개념적으로 같지만, 가중치를 계산하는 방법은 차이가 있다. 아래의 식은 인코더에서 어텐션 가중치가 계산되는 식이다.

$$e_t^k = v_e^T \tanh(W_e[h_{t-1}; s_{t-1} + U_e x^k]) \quad (2.7)$$

$$a_t^k = \frac{\exp(e_t^k)}{\sum_{i=1}^n \exp(e_t^i)} \quad (2.8)$$

위의 식 2.7에서 h_{t-1} 은 이전 시간단계의 hidden state이고, s_{t-1} 은 인코더 LSTM유닛 안에 있는 cell state이다. 그리고 $x^k = (x_1^k, x_2^k, \dots, x_T^k)$ 는 입력으로 들어가는 외생변수들이다. 어텐션 점수 e_t^k 를 계산할 때는 이전 시간단계의 hidden state와 cell state를 이어줘서 학습을 한다. Qin et al.[25]은 시간단계가 증가할수록 정보가 cell state에 누적 되기때문에 LSTM이 가지고 있는 기울기 값의 손실문제를 극복하고, 모델이 시계열 데이터가 가지고 있는 장기종속성(long-term dependencies)을 잘 파악할 수 있다고 한다.

인코더에서 사용되는 어텐션 메커니즘이 적절한 외생변수들을 선택하기 위해서 사용된다면, 디코더에서 사용된 어텐션 메커니즘은 모든 시간간격 사이에서 적절한 인코더의 hidden states를 고르는 방법이다. Qin et al.[25]은 디코더에서 사용된 어텐션 메커니즘을 temporal attention mechanism이라고 명명하였다. 디코더에서 어텐션 점수가 계산되는 방법은 다음과 같다.

$$l_t^i = v_d^T \tanh(W_d[d_{t-1}; s'_{t-1} + U_d h_i]), \quad 1 \leq i \leq T \quad (2.9)$$

$$b_t^i = \frac{\exp(l_t^i)}{\sum_{j=1}^T \exp(l_t^j)} \quad (2.10)$$

식 2.9에서 d_{t-1} 는 이전 시간단계의 hidden state이고, s'_{t-1} 는 cell state이다. 인코더에서와 마찬가지로 hidden state와 cell state를 연결지어서 어텐션 점수를 계산한다. 그리고 계산된 l_t^i 를 softmax함수를 통해서 b_t^i 어텐션의 가중치가 구해진다. 어텐션 가중치 b_t^i 는 예측에 있어서 i 번째 인코더 hidden state의 중요정도를 나타낸다. 그 다음으로 context vector c_t 는 아래와 같이 어텐션의 가중치와 모든 인코더의 hidden states들의 가중치합으로 구해진다.

$$c_t = \sum_{i=1}^T B_t^i h_i \quad (2.11)$$

그 다음은 시계열적인 요소를 반영하기 위해서 context vector와 예측변수의 이전 값을 활용해서 새로운 벡터를 계산하고 디코더의 hidden state를 업데이트한다. 자세한 방법은 다음 식과 같다. y_{t-1} 은 예측변수의 이전 값이고, f 는 비선형함수이다.

$$\tilde{y}_{t-1} = \tilde{w}^\top [y_{t-1}; c_{t-1}] + \tilde{b} \quad (2.12)$$

$$d_t = f(d_{t-1}, \tilde{y}_{t-1}) \quad (2.13)$$

본 연구에서는 Qin et al.[25]이 제안한 듀얼 어텐션 메커니즘을 활용하였다. 그러나 디코더의 hidden state가 업데이트되는 과정을 시계열 데이터가 가지고 있는 특성을 반영할 수 있도록 방법을 개선하였다. 자세한 방법은 디코더의 구조에 대해서 설명하는 3.3장에서 하겠다.

제 3 장 제안하는 방법

3.1 제안하는 Encoder-Decoder의 구조

Qin et al.[25]이 제안한 모델 DA-RNN은 식 1.1에서 나타낸 NARX 시계열 예측모델이다. 그리고 모델을 비교 및 평가하는 데이터로 주식데이터인 Nasdaq100를 사용하였다. 모델이 예측하는 값 y_t 는 t 시점의 Nasdaq100 지수값이고, 외생변수들로는 Nasdaq100에 포함되는 100개 기업의 개별종목가격이 사용되었다. DA-RNN은 타 모델들과 비교해서 우수한 성능을 보였고, 듀얼 어텐션 메커니즘을 제안한 Qin et al.[25]의 연구는 많은 주목을 받았으며 그 의의가 있다. 그러나 한 가지 비판할 점은 주가데이터를 사용한 NARX모델이 현실적으로 적용하기 힘들다. 그 이유는 t 시점의 Nasdaq100 지수 y_t 를 예측하는데 외생변수의 시점이 예측하고자 하는 시점과 같은 t 시점까지 입력으로 들어가기 때문이다. 따라서 본 연구에서는 식 1.1를 수정하여 다음과 같은 시계열 예측모델을 제안한다.

$$y_t = F(y_{t-1}, y_{t-2}, \dots, u_{t-1}, u_{t-2}, \dots) + \epsilon_t \quad (3.1)$$

위의 식 3.1은 y_{t+1} 를 예측하는데 타겟변수의 시점과 외생변수들의 시점이 모두 예측시점의 이전 시점인 t 까지만 입력으로 사용되기 때문에 현실적인 시계열 예측모델이라고 할 수 있다. 본 연구에서 제안하는 시계열 예측 Encoder-Decoder 구조를 도식화하면 아래의 그림 3.1과 같다.

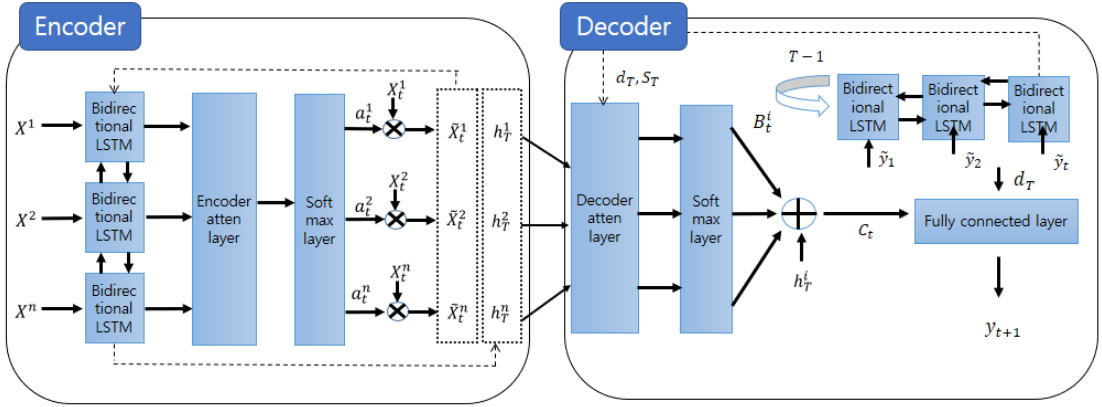


Figure 3.1: 제안하는 모델의 구조

본 연구에서 제안하는 모델은 인코더와 디코더에 양방향 LSTM 모델을 사용하였다. 양방향 LSTM은 기본 LSTM 모델보다 hidden states가 두 배 많기 때문에 학습되는 계산량은 더 복잡하다. 그러나 이전 단계에서 흐르는 정보들만 받는 LSTM과 다르게 지정한 입력값의 길이의 끝에서부터 처음까지 한 번 더 학습이 되기 때문에 더 많은 정보를 학습하고 보존할 수 있다. 시계열 데이터의 경우 데이터의 움직임을 나타내는 추세를 파악하고 학습하는 것이 중요하기 때문에 양방향 LSTM을 사용하였다.

인코더에서는 Qin et al.[25]의 연구와 같이 어텐션 메커니즘을 통해서 여러 외생변수 중에서 예측에 적합한 외생변수들을 선택하는 것이 목표이다. 그러나 Qin et al.[25]의 연구에서 context vector에 의존하여 디코더의 hidden state를 업데이트했다면, 본 연구에서는 매 시간 단계마다 context vector를 사용하지 않았다. 디코더단에 있는 양방향 LSTM에서 양방향으로 학습된 hidden states와 타겟변수의 요약통계치를 활용하여 디코더의 hidden state를 시간 단계마다 업데이트하였다. 타겟변수의 과거값들에 집중하여 학습하는 것이 타겟변수의 추세를 파악하고 다음 값을 예측하는 데 적합하기 때문이다.

3.2 Encoder의 내부구조

인코더의 구성은 듀얼 어텐션 메커니즘을 제안한 Qin et al.[25]의 연구와 크게 다르지 않으며, 구조를 자세히 설명하면 다음과 같다. 본 연구에서 제안하는 모델은 NARX를 기반으로 하는 시계열 예측모델로, 타겟변수의 다음시점의 값 y_{t+1} 를 예측하는데 여러 외생변수가 사용된다. 따라서 입력으로 들어가는 외생변수들은 다음과 같이 표시한다. $X = (x^1, x^2, \dots, x^n)^\top$, 즉 x^1 은 외생변수 하나의 시계열 데이터이고, 총 n 차원의 외생변수들이 입력으로 들어간다. 기계번역 분야에서 고정된 길이의 문장 단위로 입력이 들어가는 것처럼, 시계열 데이터 역시 고정된 길이의 입력값이 들어가야 한다. 고정된 시간간격(fixed window size)을 T 라고 하면, 하나의 외생변수를 다음과 같이 표현할 수 있다.

$$x^k = (x_1^k, x_2^k, \dots, x_T^k)^\top, 1 \leq k \leq n \quad (3.2)$$

즉, T 까지의 길이를 가진 시계열 외생변수 n 개가 양방향 LSTM층에 입력된다. 양방향 LSTM의 기본구조는 LSTM과 다르지 않지만 양방향으로 학습이 되기 때문에 hidden state의 수가 2배가 된다. 따라서 인코더단에 어텐션 점수가 계산되는 법은 다음과 같다.

$$e_t^k = v_e^\top \tanh(W_e[\vec{h}_{t-1}; \overleftarrow{h}_{t-1}; \vec{s}_{t-1}; \overleftarrow{s}_{t-1}] + U_e X^k) \quad (3.3)$$

$$a_t^k = \frac{\exp(e_t^k)}{\sum_{i=1}^n \exp(e_t^i)} \quad (3.4)$$

위 식 3.3 에서 h_{t-1} 은 이전 시간 단계의 인코더 양방향 LSTM의 hidden state이고, s_{t-1} 은 이전 시간 단계의 인코더 양방향 LSTM의 cell state이다. 양방향 LSTM을 사용하였기 때문에 hidden state와 cell state가 방향에 따라서 두 가지가 나오게 된다. 따라서 어텐션 점수 e_t^k 를 계산할 때는 식 3.3과 같이 이전 단계에서 나온 모든 hidden

state와 cell state를 연결해서 모든 k 차원의 외생변수들과의 유사도를 계산한다. 이전 단계의 hidden state와 cell state는 과거의 정보들을 담고 있으므로, 어텐션 점수 e_t^k 는 t 시점의 외생변수들의 중요도를 나타낼 수 있다. 계산된 e_t^k 를 softmax 함수를 통해서 식 3.4와 같이 계산을 하여서 어텐션 점수의 가중치 a_t^k 를 구할 수 있다.

마지막으로 계산된 가중치 a_t^k 를 해당하는 외생변수들과 곱해주면 인코딩된 외생변수값이 나오게되고, 이를 이용해서 입력값의 마지막 시점인 T 까지의 인코더 hidden state를 업데이트해 줄 수 있다. 식으로 나타내면 다음과 같다. \tilde{x}_t 는 인코딩된 외생변수들이다.

$$\tilde{x}_t = (a_t^1 x_t^1, a_t^2 x_t^2, \dots, a_t^n x_t^n)^\top \quad (3.5)$$

$$h_t = f(h_{t-1}, \tilde{x}_t) \quad (3.6)$$

이처럼 인코더의 어텐션 메커니즘을 활용하면 모델이 모든 외생변수를 동등하게 보지 않고, 가중치에 따라서 중요성을 판단할 수 있다. 인코더 hidden state의 업데이트가 끝나면, 그림 3.1에 나타난 바와 같이 디코더로 전달하게 된다.

3.3 Decoder의 내부구조

디코더에서도 인코더와 마찬가지로 양방향 LSTM을 사용하였다. 그리고 Qin et al.[]가 제안한 방법과 다르게 시계열 데이터의 특성이 학습되도록 모델을 구성하였다. 먼저 디코더에서 어텐션 점수가 구해지는 방법은 인코더에서 구해진 방법과 크게 다르지 않다. 디코더 어텐션의 가중치 B_t^i 가 계산되는 방법은 다음과 같다.

$$l_t^i = v_d^\top \tanh(W_d[\vec{d}_{t-1}; \overleftarrow{d}_{t-1}; \vec{s}'_{t-1}; \overleftarrow{s}'_{t-1}] + U_d h_{T-1}^i) \quad (3.7)$$

$$B_t^i = \frac{\exp(l_t^i)}{\sum_{j=1}^T \exp(l_t^j)} \quad (3.8)$$

인코더에서처럼 이전 시간 단계의 디코더 양방향 LSTM hidden state인 d_{t-1} 와 cell state인 s'_{t-1} 을 연결해준다. 그리고 인코딩된 외생변수들 \tilde{X}_i 를 통해서 T 시점까지 업데이트가 완료된 인코더의 hidden states와의 유사도를 계산한다. 여기서 Qin et al.[]의 연구와 큰 차이점이 있다. 본 연구에서는 context vector를 이용하여 디코더의 hidden state와 cell state를 업데이트하지 않는다. 그 이유는 시계열 데이터가 가지고 있는 특성을 더 반영하기 위해서이다. 아래의 그림은 제안하는 모델의 디코더 구조이다.

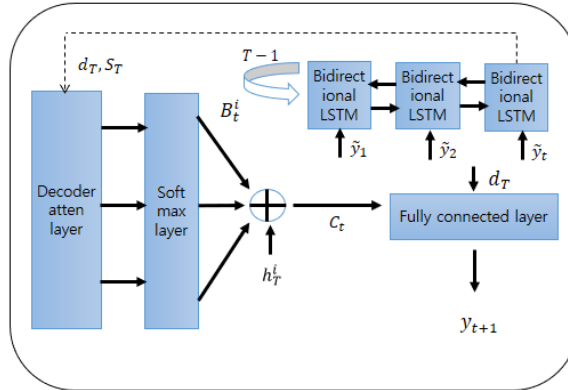


Figure 3.2: 디코더의 구조

위 그림 3.2에 나타난바와 같이 양방향 LSTM을 업데이트 시키는데 context vector가 사용되지 않는다. 오직 타겟변수 y 의 과거값들과 요약 통계치들만 디코더의 양방향 LSTM의 입력으로 들어간다. 타겟변수의 T시점까지의 과거값들과 요약 통계치정보를 다음과 같이 표현할 수 있다.

$$y' = (y_{T-t}, \dots, y_{T-1}, y_{T-t}^1, \dots, y_{T-1}^1, \dots, y_{T-1}^n) \quad (3.9)$$

$$y'_t = (y_t, y_t^1, \dots, y_t^n) \quad (3.10)$$

위 식 3.10에서 y' 은 타겟변수의 과거값들과 고정된 시간 간격 이전의 과거값들에 대한 통계치정보를 담고 있다. 위의 식에서 y^n 은 n가지의 다양한 요약 통계치정보를 의미한다. 타겟변수의 요약 통계치변수를 만들때는 feature engineering이 필요하다. 요약 통계치를 활용하는 이유는 다음과 같다. Encoder-Decoder 구조의 모델에서는 기계번역 태스크에서 문장 단위로 끊어서 학습하듯이, 시계열 데이터를 고정된 시간 간격으로 끊어서 모델의 입력으로 넣어준다. 그러나 시계열 데이터는 자연어보다 데이터 간의 연결성이 중요시된다. 따라서 고정된 시간 간격을 $T = 10$ 으로 설정하면, 매번 10의 간격으로만 학습이 되고 그 이전의 예측변수의 움직임에 대한 정보를 잃어버리게 된다. 그러므로 시간 간격 이전의 타겟변수의 추세와 흐름에 대한 정보를 반영하기 위해 요약 통계치를 활용하였다. 본 연구에서는 주가 데이터를 이용하여 모델의 성능을 평가하였고, 요약 통계치로는 주가의 추세를 나타내는 기술적 보조지표(technical indicator)를 활용하였다. 요약 통계치에 대한 자세한 설명은 4장 데이터 설명란에 있다.

디코더단에 양방향 LSTM의 입력으로 들어가는 \tilde{y} 는 다음과 같이 계산이 된다.

$$\tilde{y}_t = \tilde{w}^\top [y'_{t-1}; \overrightarrow{d}_{t-1}; \overleftarrow{d}_{t-1}] + \tilde{b} \quad (3.11)$$

이전 단계의 y' 과 양방향 hidden states를 연결하여 학습시켜준 뒤, 계산된 \hat{y} 가 양방향 LSTM의 입력으로 들어간다. \hat{y} 는 T-1 까지 업데이트가 되어 입력으로 들어가기 때문에 디코더의 hidden state도 T-1만큼 업데이트가 된다. 따라서 디코더단의 hidden state를 입력으로 받는 어텐션 가중치 B_t^i (식 3.7, 3.8)도 T-1 만큼 업데이트가 된다. 아래의 식은 context vector가 계산되는 법이다.

$$C_t = \sum_{i=1}^T B_t^i h_T^i \quad (3.12)$$

위 식과 같이 context vector는 어텐션 가중치 B_t^i 와 T시점까지 업데이트된 인코더의 hidden state의 가중치 합으로 계산이 된다. 그러므로 context vector 역시 T-1까지 업데이트된다. 마지막으로 예측이 이뤄지는 완전연결망은 아래와 같이 계산된다.

$$y_{t+1} = v_y^\top (W_y [\vec{d}_T; \overleftarrow{d}_T; C_t] + b_w) \quad (3.13)$$

식 3.11을 보면 디코더단의 양방향 LSTM의 입력으로 타겟변수에 관한 변수들만 들어가는 걸 알 수 있다. 디코더의 hidden state는 타겟변수의 과거 정보들만을 담고 있고, 식 3.7에서는 디코더의 hidden state와 업데이트된 인코더의 hidden state의 유사도 점수가 계산된다. 따라서 디코더의 어텐션 메커니즘은 시점마다 타겟변수의 추세에 관한 정보들과 외생변수들의 사이의 중요도를 나타내는 점수라고 볼 수 있다.

제 4 장 실험 결과

4.1 데이터 설명

본 연구에서 사용한 데이터는 한국거래소 종합주가지수 KOSPI 200을 추적하는 KODEX 200에 대한 데이터를 사용하였다. KOSPI 200이란, 주식시장의 산업별 최우량 200개 기업의 주가를 종합한 지수이다. KOSPI 200의 지수와 KOSPI 지수의 상관관계는 상당히 높기 때문에 신뢰도가 높은 지수라고 볼 수 있다. KOSPI 200을 구성하는 종목들이 KOSPI 시가총액의 약 80%를 차지하고 있기 때문이다. 그러나 한국거래소에서 하루의 한번 KOSPI 200 지수를 발표하기 때문에 일 단위의 데이터는 양이 충분하지 않다. 따라서 본 연구에서는 5분 단위의 KODEX 200 데이터와 구성 종목들의 개별 주가 데이터를 사용하였다. 5분 단위의 거래데이터는 대신증권이 제공하는 API ”크레온”을 통해서 수집하였다. 아래의 그림4.2은 KODEX 200 상위 8개 구성 종목의 비중이다.

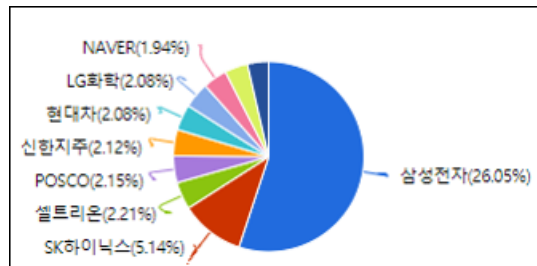


Figure 4.1: 상위 8개 구성종목의 구성비중

KODEX 200의 구성 비중은 기업의 시가총액의 순으로 계산이 된다. 위 그림 4.2에서 보다시피 삼성전자의 구성 비중이 압도적으로 높고, 보통 상위 10개 구성 종목

비중의 합이 50% 가까이 차지하고 있다. 시가총액이 낮을수록 비중이 작으며, 현재 구성 비중이 100위인 "롯데지주"가 차지하는 비중은 0.13 %이다. 따라서 본 연구에서는 KODEX 200를 예측하는데 구성 종목 중 상위 100개의 종목만을 외생변수로써 사용하였다. 사용한 데이터의 기간은 2015년 2월 2일 9시 05분부터 2019년 3월 18일 3시까지 약 4년 치의 5분 단위의 거래 데이터를 사용하였고, 약 70,000개의 데이터이다. 그리고 해당 기간 동안 거래 중지 기간이 길었던 종목이나, 2015년 이후에 상장된 종목은 제외하였다. 제외된 종목은 "삼성바이오로직스", "우리금융지주", "넷마블", "현대중공업지주", "오리온", "BGF라테일", "대우조선해양", "두산밥캣" 총 8개 종목이다. 아래의 그림 4.2은 데이터 프레임 형태로 정리한 외생변수들이다.

	삼성 전자	SK하이닉스	셀트리온	POSCO	신한 지주	현대차	LG화학	KOR 콤	NAVER	현대비 즈	한화 생명	한화 카드	현대 지주	현대 모빌	한국 에너지	한화 에너지	한국 가스	GS리 테일			
0	27360	47550	60400	268500	42500	159000	222000	37800	133200	240000			7640	75500	73200	17300	27150	24350	4475	61900	26250
1	27360	47350	60200	269000	42750	160000	224000	37450	133800	240000			7670	75400	74120	17500	27150	24500	4445	61800	26650
2	27320	47300	61800	268500	42600	159000	224000	37650	133600	239500			7690	75600	74180	17400	27150	24550	4450	61700	26800
3	27340	47350	60700	268000	42500	159000	223500	37400	133200	240000			7690	75300	73960	17450	27200	24500	4450	61400	26800
4	27360	47350	63600	267500	42550	158000	225500	37450	133200	239000			7670	75000	73960	17550	27250	24350	4495	60700	26800
5	27360	47250	65700	268000	42650	158000	224000	37400	133000	238000			7680	74900	74000	17650	27200	24400	4535	60000	26700
6	27340	47250	66600	268000	42750	158000	224000	37400	132600	238000			7690	75000	73920	17600	27200	24300	4510	60100	26700
7	27360	47350	64800	268000	42600	158000	223500	37500	134600	237500			7670	75300	73880	17600	27180	24250	4520	60400	26700
8	27380	47350	65300	267000	42700	159000	224000	37450	135000	238000			7600	75300	73720	17600	27180	24250	4550	60100	26550
9	27380	47350	65700	268000	42600	158000	223500	37450	134400	238000			7670	74900	73640	17550	27180	24250	4540	60000	26550
10	27360	47350	65200	267500	42700	158000	224000	37450	134000	237500			7680	74800	73640	17500	27150	24100	4535	60200	26300
11	27360	47350	65100	268000	42600	158000	224000	37450	134200	237500			7680	74500	73640	17500	27150	24150	4545	60100	26550
12	27380	47350	64700	268000	42600	158000	224000	37450	134000	238000			7680	74500	73600	17500	27150	24400	4550	60200	26450
13	27380	47300	64700	268000	42600	159000	223500	37400	134000	239000			7690	74000	73640	17500	27150	24350	4555	60200	26300
14	27380	47350	65200	268000	42600	159000	224000	37400	134200	238500			7680	74000	73600	17500	27200	24300	4555	60400	26250
15	27380	47350	65300	267500	42600	158500	223500	37400	134200	239000			7680	74000	73800	17500	27200	24300	4585	60100	26300
16	27380	47350	65300	267500	42550	159000	223500	37400	134200	239000			7690	74700	73760	17500	27150	24250	4605	59800	26300
17	27380	47300	65100	267500	42600	159000	224500	37450	134200	240000			7690	74800	74000	17550	27150	24250	4795	59900	26600
18	27400	47300	64700	268000	42500	159000	224500	37400	134200	239000			7690	75100	74040	17550	27200	24250	4920	59900	26600
19	27400	47400	65000	268000	42500	159500	224500	37400	134000	239000			7690	75000	74000	17500	27200	24250	4935	60000	26600
20	27440	47450	64700	268000	42550	158500	224500	37450	134000	239500			7690	74800	73920	17550	27200	24250	4860	59700	26600

Figure 4.2: 데이터의 예시

따라서 본 연구에서 사용하는 타겟변수는 KODEX 200 지수의 값이고, 사용되는 외생변수들은 KODEX 200의 구성 종목 92개의 개별종목 주가 데이터이다.

4.2 데이터 전처리 및 학습 방법

식 3.1에서 나타낸 바와 같이 본 연구에서 제안하는 모델은 t 시점까지의 외생변수들과 타겟변수의 값으로 타겟변수의 $t+1$ 값을 예측하는 모델이다. 본 연구에서 사용한 고정된 시간 간격은 $T = 9$ 이다. 연구에서는 5분 간격의 거래 데이터를 사용하였으므로, 매번 과거 45분의 주가 데이터를 이용해서 50분째의 타겟변수 KODEX 200의 값을 예측하는 문제가 된다. 외생변수들로 사용한 92개의 개별종목의 주가마다 크기가 다르므로 정규화가 필요하다. 모델이 배치학습에 의해서 학습이 되므로 이를 응용하여 정규화를 진행하였다. 식으로 표현하면 다음과 같다.

$$batch_norm(x_i) = \frac{x_i - mean(x_{i-9}, ..., x_i)}{std(x_{i-9}, ..., x_i)} \quad (4.1)$$

배치학습을 이용해서 모델을 학습하므로 식 4.1과 같은 방법으로 모든 데이터에 대해서 과거 9일 치를 이용한 정규화를 진행하였다. 타겟변수인 KODEX 200도 위와 같은 방법으로 정규화를 시켰다.

디코더의 내부구조를 설명하는 3.3장 식 3.9에서 요약 통계치 변수들을 활용하면 고정 시간 간격 이전의 정보를 학습할 수 있다고 하였다. 본 연구에서는 주가 데이터를 사용하였으므로, 요약 통계치 변수들로 기술적 지표들을 사용하였다. 기술적 지표는 시계열 데이터의 추세나 변동성 등을 통계적인 접근법을 이용하여 나타낸 지표이다. 본 연구에서 사용한 기술적 지표는 총 4가지의 변수들이고, 식 3.10과 같이 표현하면 아래와 같다.

$$y' = (y_{T-t}, ..., y_T, y_{T-t}^1, ..., y_T^1, ..., y_{T-t}^4, ..., y_T^4) \quad (4.2)$$

타겟변수의 과거값을 포함하여 총 5차원의 변수가 디코더단에 양방향 LSTM의 입력으로 사용이 된다. 아래의 표 4.1은 사용한 기술적 지표를 특성에 따라서 정리한 표이다.

Table 4.1: 사용한 요약 통계량 변수

특성	방법
Volatility	Bollinger Bands Indicators (Lower/upper)
Trend	Moving Average Convergence Divergence
Momentum	Stochastic Oscillator

위 테이블 4.1에서 사용한 지표들은 특성별로 가장 많이 사용되는 방법들이다. 주가의 Volatility를 나타내는 지표 Bollinger Bands는 다음과 같이 계산된다.

$$\text{Upper_Bollinger_Band} = MA(TP, n) + m * \sigma[TP, n] \quad (4.3)$$

$$\text{Lower_Bollinger_Band} = MA(TP, n) - m * \sigma[TP, n] \quad (4.4)$$

위 식에서 MA = 이동평균법, TP 는 typical price로 다음과 같이 계산된다.

$$TP = (High + Low + Close) \div 3 \quad (4.5)$$

식 4.3, 4.4에서 n 은 이동평균법이 계산되는 n 시점이고, m 은 TP 의 n 시점까지의 계산된 표준편차 σ 를 설정하는 값이다. 본 연구에서는 $n = 25$, $m = 2$ 를 사용하였다.

기술적인 지표들을 이용해 주가의 움직임을 분석하는 차트분석에서 Bollinger Bands는 매우 인기있는 지표이다. Upper_Bollinger_Band와 Lower_Bollinger_Band를 이용해 주가의 현재값이 과매수 상태인지 아니면 과매도 되었는지 판단을 할 수 있다. 아래의 그림 4.3은 Bollinger Bands의 사용 예시이다.



Figure 4.3: Bollinger Bands의 사용 예시

주가가 Bollinger Bands의 아랫선에 위치하면 과매도 된 상태로, 정규분포를 가정하였을 때 선 아래로 벗어날 확률은 2 표준편차이다. 마찬가지로 주가가 Bollinger Bands의 위에 선에 위치하면 과매수 된 상태로 주가가 선위로 더 올라갈 확률 역시 2 표준편차이다. 이처럼 기술적 지표를 활용하여 시간 간격 이전의 정보들을 반영하였다. 모델을 학습할 때 배치학습과 Adam optimizer[19]를 활용하였다. 전체 데이터 중에서 70%를 학습 데이터로, 30%를 평가 데이터로 사용하였다. 따라서 학습 데이터는 약 41,000개 그리고 평가 데이터는 약 21,000개이다. 평가방법으로 평균 제곱근 오차(MSE)를 사용하였고, 학습률은 0.001로 설정하였다. 모델의 손실함수는 다음과 같다.

$$Loss_func = \frac{1}{N} \sum_{i=1}^N (\hat{y}_T^i - y_T^i)^2 \quad (4.6)$$

배치사이즈는 128로 설정하였다. 그러므로 한 배치에 인코더에는 $(128 \times 9 \times 92)$ 차원의 텐서가 디코더에는 $(128 \times 9 \times 5)$ 차원의 텐서가 입력으로 들어간다. 본 연구를 수행하는데 사용된 프로그래밍 언어는 PyTorch이다.

4.3 성능 평가 및 비교

모델간의 성능 평가를 할 때 직관적인 기준을 추가하기 위해서 "Accuracy" 평가 지표를 추가하였다. 모델이 예측한 타겟변수의 다음 값 \hat{y}_{t+1} 과 실제 다음 값 y_{t+1} 을 비교하여 주가의 상승과 하락 방향성을 맞추는지 평가하는 방법이다. 만약 실제 다음 값의 변화가 없어서 상승과 하락을 판단할 수 없는 경우는 제외하였다. 아래는 "Accuracy"를 계산하는 방법이다.

$$y_T - y_{T-1} > 0 \text{ and } \hat{y}_T - y_{T-1} > 0 \Rightarrow \text{Correct}$$

$$y_T - y_{T-1} < 0 \text{ and } \hat{y}_T - y_{T-1} < 0 \Rightarrow \text{Correct}$$

또한, 평가 데이터에서 MSE를 계산할 때 식 4.1과 같이 정규화 하였던 값들을 원래의 값으로 변환하여서 계산하였다. 비교하는 각 모델별로 인코더, 디코더의 양방향 LSTM의 hidden state 사이즈를 64, 128으로 나눠서 실험을 진행하였고, 각 모델별로 Epoch을 최대 100번까지 학습을 하면서 평가 데이터에 대한 성능이 가장 좋은 경우를 모델의 성능으로 선택하였다.

아래의 표 4.2은 모델별로 성능을 정리한 표이다. 기준점이 되는 모델은 Encoder-Decoder 모델이다. 그리고 듀얼 어텐션 메커니즘을 제안한 Qin et al.[25]의 DA-RNN 모델과의 비교가 중점이 되었다. 아래의 4개 모델은 본 연구에서 제안하는 모델이다. 그리고 아래 2개의 모델은 기술적 지표를 요약 통계치 변수로써 사용한 모델이다. 다른 모델들과 공정한 성능 비교를 위해서 요약 통계치 변수를 사용하지 않은 모델의 성능도 포함하였다. 표 4.2에서 보는 바와 같이 요약 통계치를 사용한 모델들이 Accuracy와 MSE 모든 평가 지표에서 가장 좋은 성능을 보였다. 양방향 LSTM의 hidden state 사이즈가 64일 때 Accuracy가 "69.83%"으로 제일 높았고, MSE의 경우에는 hidden

Table 4.2: 성능 평가 결과표

Model	Accuracy	MSE
Encoder-Decoder (64) [2014]	61.77 %	1688.0264
Encoder-Decoder (128) [2014]	62.34 %	1601.6154
DA-RNN (64) [2017]	66.65 %	1271.1487
DA-RNN (128) [2017]	66.81 %	1189.5133
DA-Bi-RNN (64) [Proposed method]	68.21 %	1129.9011
DA-Bi-RNN (128) [Proposed method]	68.02 %	1108.2297
DA-Bi-RNN with y' (64) [Proposed method]	69.83 %	1021.4859
DA-Bi-RNN with y' (128) [Proposed method]	68.79 %	1013.1352

state의 사이즈가 128일 때 가장 낮았다. 따라서 고정 시간 간격 이전의 정보를 담고 있는 요약 통계치 변수를 사용한다면, 예측에 도움이 된다는 사실을 확인할 수 있다. 요약 통계치를 사용하지 않은 모델도 DA-RNN 모델보다 모든 부분에 있어서 성능이 좋았다. 그러므로, 본 연구에서 제안하는 모델의 구조가 시계열적인 특성을 반영하였기 때문에 타겟변수의 다음 시점의 값을 예측하는데 더 효과적이었다고 해석할 수 있다.

다음 실험으로는 타겟변수인 KODEX 200만을 이용한 예측을 진행하였다. 코스피 지수는 해당 종목들의 시가총액을 기준으로 가중되어서 산출되기 때문에 코스피 지수값에 해당 종목들의 값이 반영되어 있기 때문이다. 따라서 각 개별종목주가인 외생변수를 이용하여 KODEX 200 값을 예측하는 테스트가 의미가 있는지 검증하고자 한다. 단변량 시계열 분석이기때문에 어텐션 메카니즘이 사용되는 모델은 사용하지 않았다. 실험에 사용된 모델은 ARIMA와 LSTM 두 가지 모델이다.

아래 테이블 4.4에서 보는바와 같이 타겟변수만을 사용한 모델의 경우 MSE는 전반적으로 낮은 수치를 보였지만, 다음 시점의 주가 방향성을 맞추는 "Accuracy" 평가에서는 성능이 떨어지는 것을 확인할 수 있다. 그 이유는 타겟변수만을 활용하여 모델링을

Table 4.3: 타겟변수만을 이용한 예측 모델의 성능

Model	Accuracy	MSE
ARIMA	53.38 %	1084.0591
LSTM	54.81 %	1122.4977

하는 경우 모델이 타겟변수의 직전값에 맞춰서 학습이 되기때문이다. 그러므로 다음 시점의 예측값의 이전 시점의 값과 크게 다르지 않게 예측이 되기 때문에 MSE가 낮게 나온것으로 해석할 수 있다. 또한, 본 연구에서 사용한 데이터는 5분 단위의 거래 데이터이기 때문에 타겟변수의 변화량이 상대적으로 크지 않다.



Figure 4.4: 타겟변수만을 활용한 LSTM예측모델

위 그림 4.5을 보면 모델의 예측이 잘 된 것처럼 보이지만, 자세히 보면 직전 타겟 변수 값에서 크게 벗어나지 않는 범위에서만 예측하는 것을 확인할 수 있다. 그러므로, 타겟변수만을 활용한 예측모델은 적합하지 않다고 판단할 수 있다.

4.4 어텐션 가중치 분석

어텐션 메커니즘의 장점 중 하나는 어텐션 가중치의 시각화를 통해서 모델이 해석력을 가질 수 있다. 먼저 인코더에서 사용된 어텐션 메커니즘이 제대로 작동하였는지 확인하기 위해 랜덤 노이즈 데이터를 추가하였다. 실험에 사용된 외생변수들 92개 종류의 개별 주가데이터가 0에서 1사이로 정규화가 되어있으므로, 노이즈 역시 0에서 1사이의 소수 값을 랜덤으로 생성하여 외생변수(노이즈)로 추가하였다. 결과는 아래의 그림 4.5과 같다.

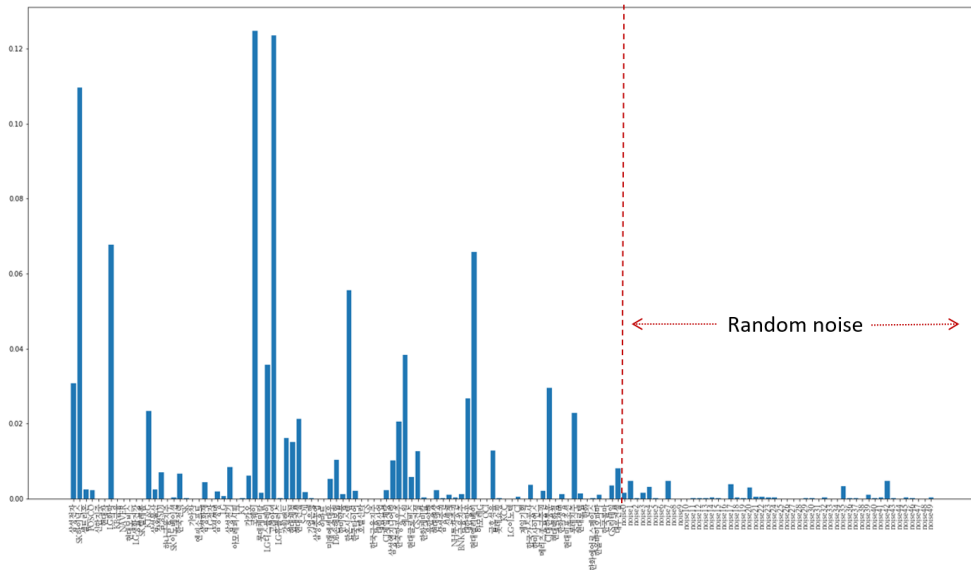


Figure 4.5: 어텐션 가중치와 랜덤 노이즈의 비교

위 그림 4.5에서 x축은 외생변수로서 사용된 KODEX 200의 구성종목 92개이다. 그리고 구성비중이 높은 순서대로 정렬을 하였다. 첫번째 변수는 삼성전자, 그 다음은 하이닉스로 시가총액 순서대로 진행된다. 빨간색 점선 이후의 변수들은 랜덤으로 생성한 노이즈 값들이다. y축은 어텐션 가중치 값이다. 그림 4.5에서 보이는 것처럼 랜덤 노이즈들의 어텐션 가중치가 상당히 낮고 대부분의 값들이 0.00에 가깝다. 위 실험 결

과를 통하여 모델이 외생변수들과 랜덤 노이즈를 구분하여 중요도를 판단한다는 것을 확인할 수 있다.

다음 실험은 KODEX 200을 구성하는 200개의 모든 종목들을 포함하여 실험을 진행하였다. 100개의 종목을 사용하여 실험하였을때와 마찬가지로 분석에 사용되는 해당기간동안 거래중지 기간이 길었거나, 2015년 이후에 상장된 종목을 제거하였다. 제외된 종목들은 "LIG넥스원", "한진중공업", "이노션", "SK케미칼", "효성중공업", "오리온홀딩스", "고려제강", "한국셀석유", "영풍", "태광산업", "남양유업", "효성", "롯데지주", "롯데칠성", "SK디스커버리", "HDC", "BGF", "쿠쿠홀딩스" 총 18개의 종목이다. 결과는 아래의 그림 4.6과 같다.

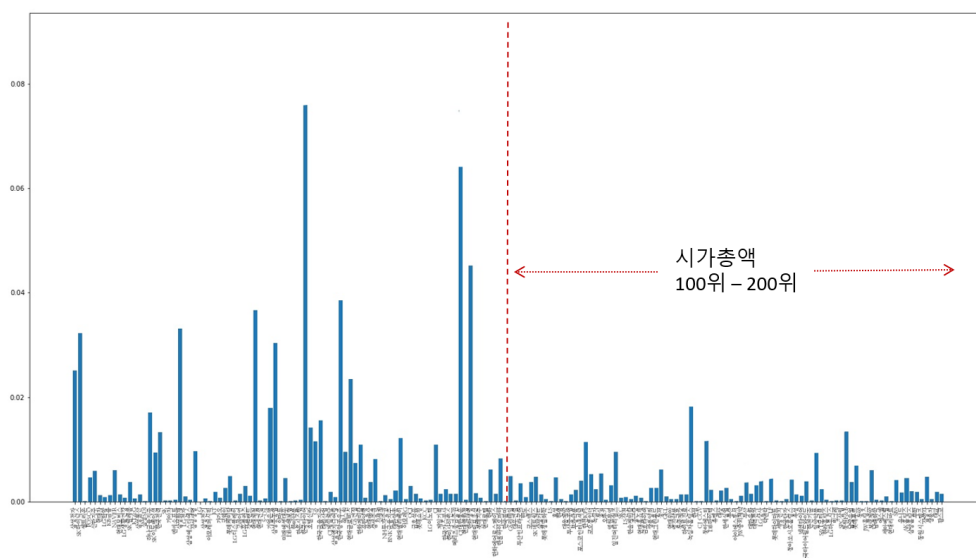


Figure 4.6: KODEX 200 모든 구성종목의 어텐션 가중치

이전 실험의 결과 그림 4.5과 마찬가지로 x축은 외생변수들로 사용된 개별 주가 종목이고, 구성비중이 큰 순서대로 정렬하였다. 그리고 빨간색 점선 이후에 나열된 종목들은 시가총액 100위에서 200위까지의 종목들로 KODEX 200의 구성비중에서 차지하는 비중이 매우 낮다. 그림 4.5에서 확인할 수 있듯이 대체적으로 시가총액 상위 100개의

종목들의 가중치가 100위 이하 종목들 보다 높다. 그러나 y 축의 가중치 값이 최대 0.08도 안된다. 모델이 선택해야하는 외생변수들이 많고, 가중치가 softmax 함수를 통해서 계산이 되므로 외생변수들의 차원이 커질수록 가중치가 대체적으로 낮아진다고 판단할 수 있다. 실제로 200위까지의 종목들을 포함하여 실험을 하였을때 MSE가 1600에서 1800사이의 값이 나왔다. 따라서 외생변수를 많이 사용한다고해서 절대적으로 좋은 성능을 보이지 않는다는 사실을 확인하였다.

어텐션 가중치 분석을 통해서 알수있는 또 한가지 사실은 KODEX 200 구성비중에서 차지하는 비중이 높다고 가중치가 높게 나오지 않는다는 점이다. 아래의 표 4.4는 평가 데이터기간 KODEX 200의 가격과 상관관계가 높은 상위 10개 종목이다.

Table 4.4: 상관계수 상위 10개 종목

종목명	상관계수	순위	종목명	상관계수	순위
삼성전자	0.6175	1	삼성증권	0.3709	65
SK하이닉스	0.4701	2	한미사이언스	0.3446	84
미래에셋대우	0.4310	47	삼성SDI	0.3403	12
한국금융지주	0.3886	50	메리츠종금증권	0.3255	81
LG화학	0.3721	8	POSCO	0.3247	6

KODEX 200 구성비중의 26%를 차지하는 "삼성전자"가 가장 상관관계가 높았고, "하이닉스", "LG화학", "POSCO"처럼 시가총액이 높은 종목들이 역시 상관계수가 높았던 반면에 "한미사이언스", "메리츠종금증권" 처럼 시가총액 순위가 높지 않음에도 불구하고 상관계수가 높은 종목도 있었다. 그러나 위 테이블 4.4은 평가데이터 기간 동안의 상관관계이고, 각 예측시점마다 나오는 어텐션의 가중치는 매번 다르다. 아래의 그림 4.7은 $t+1 = 20,653$ 번째 값을 예측하는데 모델이 판단한 외생변수들의 중요도이다.

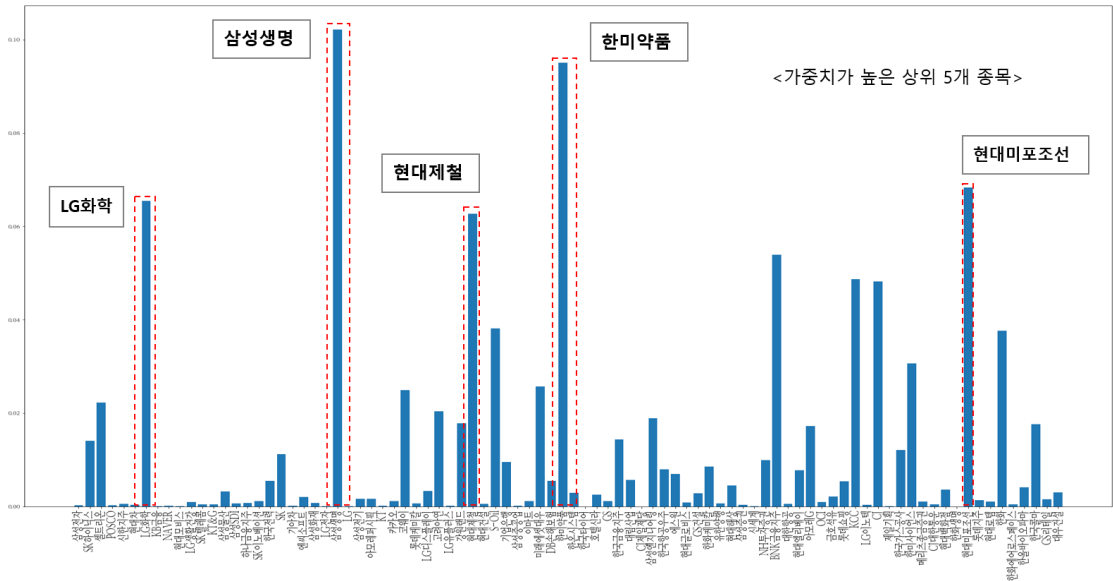
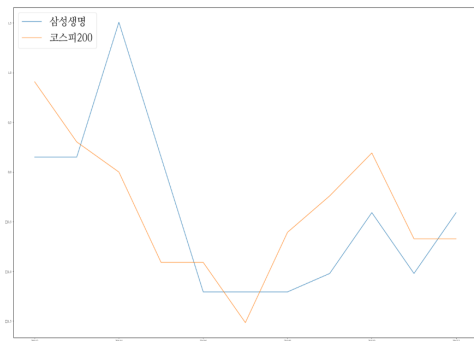
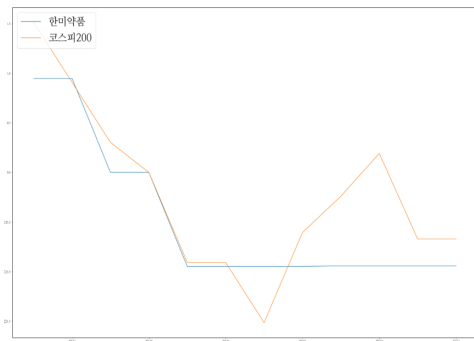


Figure 4.7: 어텐션 가중치가 높은 상위 5개 종목(20,653번째 예측값)

예시로 20,653번째의 어텐션 가중치를 사용한 이유는 일반적으로 "삼성전자"의 가중치는 대부분 높게 나오지만, 위의 경우에는 삼성전자의 가중치가 낮게 나왔기 때문이다. 그림에서 보다시피 어텐션 가중치가 높게 나온 상위 5개 종목은 "삼성생명", "한미약품", "LG화학", "현대제철", "현대미포조선" 순이다. 다음 그림은 모델이 중요하게 판단한 위 5개 종목들과 KODEX 200의 움직임을 비교하였다.

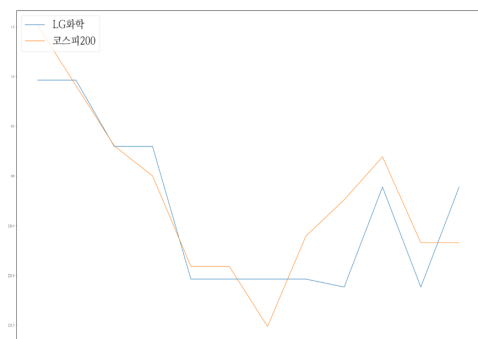


(a) 삼성생명

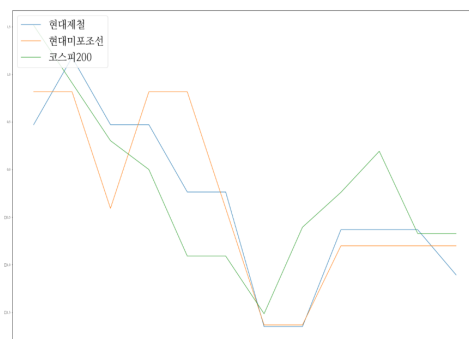


(b) 한미약품

위 그림은 20,653번째 값을 예측하는데 사용된 이전 45분($T=9$) 동안의 KODEX 200과 "삼성생명", "한미약품"의 주가 움직임이다. 노란색 선은 KODEX 200이고, 파란색 선은 해당종목이다. 모델이 예측을 하는데 중요하게 판단한 종목들과 KODEX 200의 움직임이 전체적으로 비슷하게 움직이고 있는것을 확인할 수 있다.



(a) LG화학



(b) 현대제철, 현대미포조선

다음으로 "LG화학", "현대제철", "현대미포조선"과 KODEX 200의 움직임을 보여주는 그림이다. 위에서 보았던 "삼성생명", "한미약품"의 경우처럼 어텐션 가중치가 높게 나타난 종목들은 해당 기간동안 KODEX 200과 움직임이 유사한 것을 확인할 수 있다. "현대제철"과 "현대미포조선"은 같은 산업군으로 주가의 움직임이 비슷하게 움직이는 경향이 있다. 두 종목의 가중치가 모두 높게 나왔으므로, 모델이 유사 산업군의 주가 움직임에 대해서도 잘 판단할 수 있는 것을 확인할 수 있다.

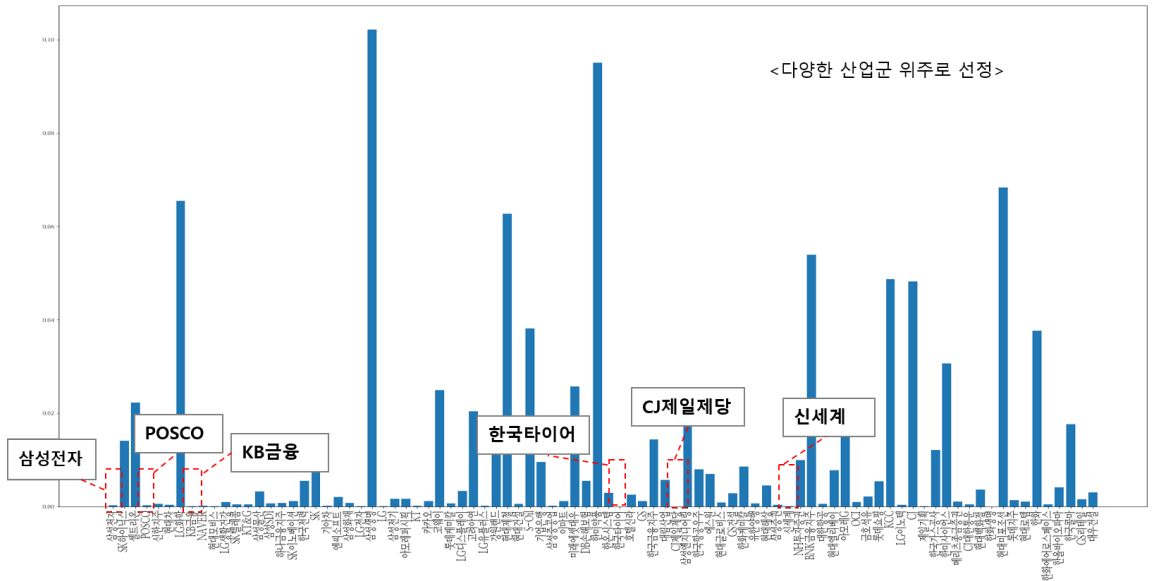
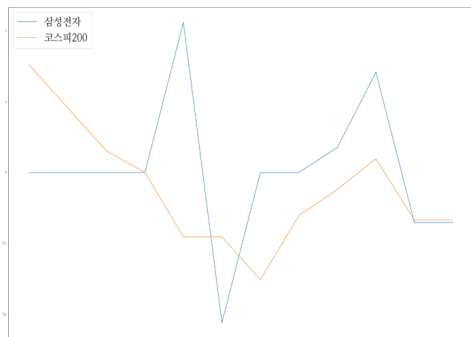
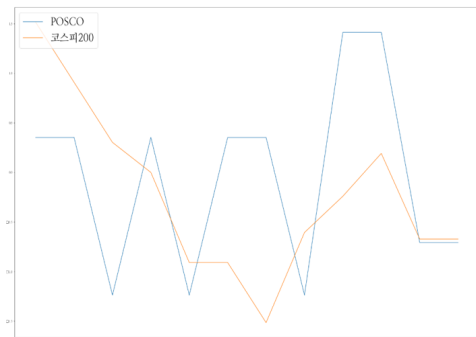


Figure 4.10: 어텐션 가중치가 낮은 하위 7개 종목(20,653번째 예측값)

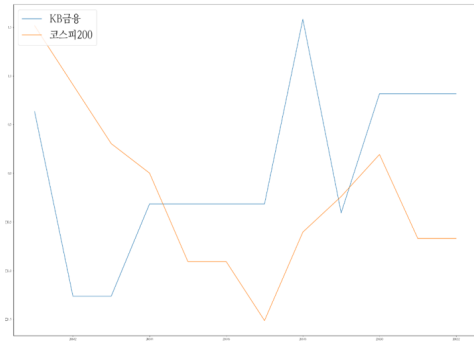
다음 실험은 20,653번째 값을 예측하는데 모델이 중요하지 않다고 판단한 종목들이다. 어텐션 가중치가 낮은 종목들이 많으므로, 다양한 산업군위주로 선정하였다. 선정한 6개의 종목은 다음과 같다. "삼성전자", "POSCO", "KB금융", "한국타이어", "CJ제일제당", "신세계"이다. 위의 그림 4.10은 해당종목들의 가중치의 값이고, 아래는 각 종목들과 KODEX 200의 움직임을 비교하는 그림들이다.



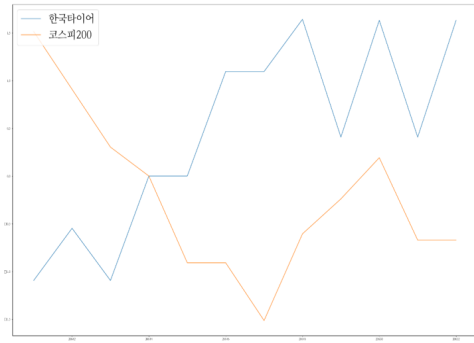
(a) 삼성전자



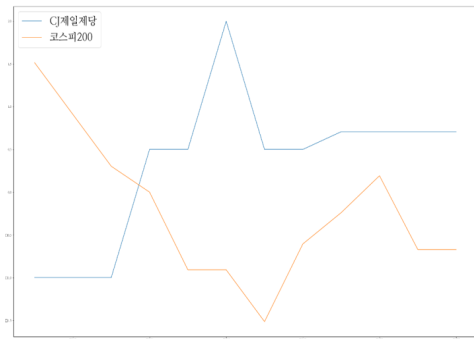
(b) POSCO



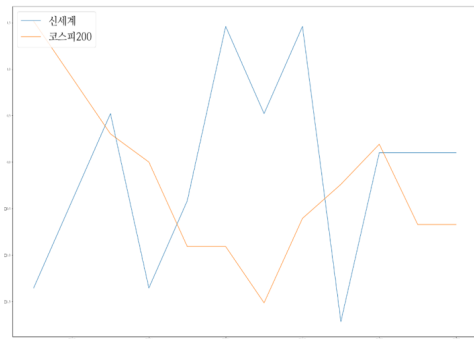
(a) KB금융



(b) 한국타이어



(c) CJ제일제당



(d) 신세계

위 그림들은 어텐션 가중치가 낮은 6개 종목들과 KODEX 200의 움직임을 나타낸다. 그림에서 보다시피 "삼성전자"를 제외하고 모든 종목들과 KODEX 200의 움직임이 반대방향으로 움직이고 있는것을 확인할 수 있다. 위 6개의 종목들 모두 모델이 예측하는데 도움이 되지 않는다고 판단한 종목들이다. "삼성전자"의 경우, 최근 25분의 주가 움직임은 KODEX 200과 비슷하게 움직이지만, 40분전의 "삼성전자"의 주가가 급격하게 튀었다가 내려와서 모델이 제대로 판단하지 못 한 경우인 것 같다. 그러나 전체적으로 모델이 중요도를 낮게 판단한 종목들은 KODEX 200과 반대로 움직이고, 중요도를 높게 준 종목들은 같은 방향으로 향하고 있음을 확인하였다. 따라서 듀얼 어텐션 메커니즘을 활용한 모델들이 성능 평가에서 다음 예측 값의 상승과 하락을 맞추는 "Accuracy" 평가에서 좋은 성능을 나타낼 수 있었다고 판단한다.

제 5 장 결론

본 연구에서는 Qin et al.[25]이 제안한 듀얼 어텐션 메커니즘의 개념을 채용하여서 시계열 데이터의 특성을 반영한 새로운 Encoder-Decoder 구조의 모델을 제안하였다. 일반적으로 어텐션 메커니즘에서는 context vector를 기반으로 모델이 담고 있는 정보를 업데이트 시킨다. 그러나 제안하는 모델에서는 디코더단에서 타겟변수를 기반으로 학습이 이루어지기 때문에 타겟변수가 가지고 있는 정보들을 우선적으로 사용한다. 따라서 타겟변수의 다음 시점의 값 y_{t+1} 을 예측하는데 더 효과적인 구조라고 할 수 있다. 모델의 성능도 Qin et al.[25]이 제안한 DA-RNN 모델보다 주가의 방향성을 맞추는 "Accuracy"와 종합적인 성능을 나타내는 "MSE"에서 보다 좋은 성능을 보였다. 또한, 기본적으로 Encoder-Decoder 구조의 모델을 사용하면 고정 시간 간격을 설정하여야 한다. 하지만 시계열 데이터의 경우 과거로부터 이어져오는 추세와 변동성이 중요하기 때문에 시간 간격으로 끊어서 학습을 하는 방법은 모든 정보를 효과적으로 학습할 수 없다. 본 연구에서는 이러한 문제점에 대응하기 위해서 통계적인 기법으로 주가의 흐름을 추적하는 기술적 지표를 예측변수의 요약 통계량으로 사용하였다. 그 결과 요약 통계치를 추가 변수로 사용하였을때 모델이 좋은 성능을 나타냄을 확인하였다. 하지만 제안하는 모델은 양방향 LSTM을 사용하였으므로, 학습되는 매개변수가 많아 모델의 복잡도가 높은 단점이 있다.

어텐션 가중치 분석을 통해서 제안하는 모델이 외생변수들과 노이즈를 구분함을 확인하였고, KODEX 200을 구성하는 모든 종목들을(외생변수 200개) 포함하였을 때 하위 100개의 종목들의 중요도가 떨어지는 사실도 확인하였다. 이는 KODEX 200의 구성 비중에서 하위 100개의 종목들이 차지하는 비중이 매우 작기 때문에 나타난 현상

이라고 판단한다. 그리고 외생변수 200개를 모두 사용하였을때 모델의 성능(MSE)도 전체적으로 하락하는 사실을 확인하였다. 또한, 모델이 매 시점 예측을 할 때 중요하다고 판단하는 종목들과 KODEX 200의 움직임을 정밀히 분석한 결과, 높은 어텐션 가중치와 실제 주가의 움직임이 상당히 유사함을 확인하였다. 마찬가지로 모델이 중요하지 않다고 판단하는 종목들은 타겟변수와 반대의 움직임을 가지고 있다. 이처럼 제안하는 모델이 어텐션 가중치를 통해서 외생변수들의 움직임과 방향성에 정보를 가지고 있기 때문에 타겟변수만을 활용한 예측모델보다 다음 시점의 주가 방향성을 맞추는데 뛰어난 결과를 보였다고 판단한다. 마지막으로 본 연구의 한계점은 주가데이터만을 사용한점이라고 볼 수 있다. 여러 외생변수들을 사용하였지만, 주가의 움직임만을 가지고 다음 시점의 특정 주가의 가격을 예측하는 테스트는 어렵다고 사료된다. 따라서 외생변수들과 타겟변수의 관계가 명확한 시계열 데이터를 활용한 추가적인 분석이 진행된다면 더 유의미한 결과를 얻을 수 있을 것이라고 판단된다.

참고 문헌

- [1] D. ASTERIOU AND S. HALL, *ARIMA Models and the Box-Jenkins Methodology*, 01 2016, pp. 275–296.
- [2] D. BAHDANAU, K. CHO, AND Y. BENGIO, *Neural machine translation by jointly learning to align and translate*, arXiv preprint arXiv:1409.0473, (2014).
- [3] Y. BENGIO, P. SIMARD, P. FRASCONI, ET AL., *Learning long-term dependencies with gradient descent is difficult*, IEEE transactions on neural networks, 5 (1994), pp. 157–166.
- [4] A. BOUCHACHIA AND S. BOUCHACHIA, *Ensemble learning for time series prediction*.
- [5] P. CHAKRABORTY, M. MARWAH, M. ARLITT, AND N. RAMAKRISHNAN, *Fine-grained photovoltaic output prediction using a bayesian ensemble*, in Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012.
- [6] S. CHEN, X. WANG, AND C. J. HARRIS, *Narx-based nonlinear system identification using orthogonal least squares basis hunting*, IEEE Transactions on Control Systems Technology, 16 (2007), pp. 78–84.
- [7] K. CHO, B. VAN MERRIËNBOER, D. BAHDANAU, AND Y. BENGIO, *On the properties of neural machine translation: Encoder-decoder approaches*, arXiv preprint arXiv:1409.1259, (2014).

- [8] K. CHO, B. VAN MERRIËNBOER, C. GULCEHRE, D. BAHDANAU, F. BOUGARES, H. SCHWENK, AND Y. BENGIO, *Learning phrase representations using rnn encoder-decoder for statistical machine translation*, arXiv preprint arXiv:1406.1078, (2014).
- [9] J. CHUNG, C. GULCEHRE, K. CHO, AND Y. BENGIO, *Empirical evaluation of gated recurrent neural networks on sequence modeling*, arXiv preprint arXiv:1412.3555, (2014).
- [10] J. DEVLIN, M.-W. CHANG, K. LEE, AND K. TOUTANOVA, *Bert: Pre-training of deep bidirectional transformers for language understanding*, arXiv preprint arXiv:1810.04805, (2018).
- [11] E. DIACONESCU, *The use of narx neural networks to predict chaotic time series*.
- [12] R. FRIGOLA AND C. E. RASMUSSEN, *Integrated pre-processing for bayesian nonlinear system identification with gaussian processes*, in 52nd IEEE Conference on Decision and Control, IEEE, 2013, pp. 5371–5376.
- [13] Y. GAO AND M. J. ER, *Narmax time series model prediction: feedforward and recurrent fuzzy neural network approaches*, Fuzzy sets and systems, 150 (2005), pp. 331–350.
- [14] A. GRAVES, A.-R. MOHAMED, AND G. HINTON, *Speech recognition with deep recurrent neural networks*, in 2013 IEEE international conference on acoustics, speech and signal processing, IEEE, 2013, pp. 6645–6649.

- [15] S. HOCHREITER AND J. SCHMIDHUBER, *Long short-term memory*, Neural computation, 9 (1997), pp. 1735–1780.
- [16] S. HOCHREITER AND J. SCHMIDHUBER, *Lstm can solve hard long time lag problems*, in Advances in neural information processing systems, 1997, pp. 473–479.
- [17] S. IOFFE AND C. SZEGEDY, *Batch normalization: Accelerating deep network training by reducing internal covariate shift*, arXiv preprint arXiv:1502.03167, (2015).
- [18] A. KARPATHY AND L. FEI-FEI, *Deep visual-semantic alignments for generating image descriptions*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3128–3137.
- [19] D. P. KINGMA AND J. BA, *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980, (2014).
- [20] Y. LIANG, S. KE, J. ZHANG, X. YI, AND Y. ZHENG, *Geoman: Multi-level attention networks for geo-sensory time series prediction*.
- [21] T. LIN, B. G. HORNE, P. TINO, AND C. L. GILES, *Learning long-term dependencies in narx recurrent neural networks*, IEEE Transactions on Neural Networks, 7 (1996), pp. 1329–1338.
- [22] Z. LIU AND M. HAUSKRECHT, *A regularized linear dynamical system framework for multivariate time series analysis*, in Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.

- [23] T. MIKOLOV, I. SUTSKEVER, K. CHEN, G. S. CORRADO, AND J. DEAN, *Distributed representations of words and phrases and their compositionality*, in Advances in neural information processing systems, 2013, pp. 3111–3119.
- [24] V. NAIR AND G. E. HINTON, *Rectified linear units improve restricted boltzmann machines*, in Proceedings of the 27th international conference on machine learning (ICML-10), 2010, pp. 807–814.
- [25] Y. QIN, D. SONG, H. CHEN, W. CHENG, G. JIANG, AND G. COTTRELL, *A dual-stage attention-based recurrent neural network for time series prediction*, arXiv preprint arXiv:1704.02971, (2017).
- [26] H. SONG, D. RAJAN, J. J. THIAGARAJAN, AND A. SPANIAS, *Attend and diagnose: Clinical time series analysis using attention models*, in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
- [27] N. SRIVASTAVA, G. HINTON, A. KRIZHEVSKY, I. SUTSKEVER, AND R. SALAKHUTDINOV, *Dropout: a simple way to prevent neural networks from overfitting*, The Journal of Machine Learning Research, 15 (2014), pp. 1929–1958.
- [28] Y. TANG, J. XU, K. MATSUMOTO, AND C. ONO, *Sequence-to-sequence model with attention for time series classification*, in 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), IEEE, 2016, pp. 503–510.
- [29] A. VASWANI, N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. N. GOMEZ, L. KAISER, AND I. POLOSUKHIN, *Attention is all you need*, in Advances in neural information processing systems, 2017, pp. 5998–6008.

- [30] P. WHITTLE, *Tests of fit in time series*, Biometrika, 39 (1952), pp. 309–318.
- [31] Y. WU, J. M. HERNÁNDEZ-LOBATO, AND Z. GHAHRAMANI, *Dynamic covariance models for multivariate financial time series*, arXiv preprint arXiv:1305.4268, (2013).

Abstract

Bidirectional Encoder-Decoder with Dual-Stage Attention for Multivariate Time-Series Prediction

Hyeonguk Lee

Department of Industrial Engineering

The Graduate School

Seoul National University

Recurrent neural network has been widely applied for time-series prediction. However, the vanishing gradient is still a problem and only few of them select the relevant dependent variables appropriately. In this paper, I propose a bidirectional encoder-decoder model using dual-stage attention to address above problems. In the encoder, input attention mechanism extracts relevant dependent variables by referring the hidden state and cell state from the bidirectional LSTM of previous time step. In the decoder, attention mechanism is applied on the past values of independent variable but it works differently with the first stage (encoder). A bidirectional LSTM runs through until the defined time step and the hidden state in the decoder is updated at each time step. The updated hidden state combines with the encoded input are used as input in the decoder. With the proposed method, the decoder

can capture the information throughout the encoder. It learns a trend of independent variable efficiently and can make better prediction in comparison with other encoder-decoder models. For the evaluation, Korean stock market 5-minute trading data is used.

Keywords: Encoder-Decoder, Attention Mechanism, Recurrent Neural Network, Time-Series Prediction, Finance

Student Number: 2017-22550